



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|-----------------|-------------|----------------------|---------------------|------------------|
| 09/518,492      | 03/03/2000  | Ram Kudukoli         | 5150-37301          | 7614             |

7590

01/13/2005

Jeffrey C. Hood  
Conley, Rose & Tayon, P.C.  
P.O. Box 398  
Austin, TX 78767-0398

|          |
|----------|
| EXAMINER |
|----------|

VU, KIEU D

|          |              |
|----------|--------------|
| ART UNIT | PAPER NUMBER |
|----------|--------------|

2173

DATE MAILED: 01/13/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

09/518,492

Applicant(s)

KUDUKOLI ET AL.

Examiner

Kieu D Vu

Art Unit

2173

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 01 November 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 285-381 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 285-381 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_.
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_.

**DETAILED ACTION**

1. Claims 285-381 are pending.

***Claim Rejections - 35 USC § 103***

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 370-380 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sojoodi et al ("Sojoodi ", USP 6437805) and Sullivan (USP 6453464)

Regarding claims 370, Sojoodi teaches a system for programmatically creating a graphical program comprising a computer system including a CPU (200) and memory (204 and 206), a client program executing in the computer system, the client program performs API calls to programmatically create a graphical program (col 5, lines 28-30), a server program operable to receive the client program calls to programmatically create a graphical program and operable to perform the respective operations (col 4, lines 43-53). Sojoodi does not teach creating the new program from a program without any user input specifying the nodes and interconnection of the plurality of node. However, such feature is known in the art as taught by Sullivan. Sullivan teaches steps for creating new program from a program without user input (converting a COBOL program to a Java program without user input (col 2, lines 8-27). Thus, it would have been obvious to one of ordinary skill in the art to apply Sullivan teaching of converting one program to another program without user's inputs in Sojoodi graphical program environment with

Art Unit: 2173

the motivation being to automatically translate graphical program from one programming language to another programming language.

Regarding claim 371, Sojoodi teaches that the server program executes on another computer system which is connected to said computer system via a network (Fig. 1A).

Regarding claims 372-373, Sojoodi teaches the creating a graphical program by obtaining a reference to a software component (col 58, lines 42-44) and invoking methods of the software component (col 6, lines 15-19).

Regarding claim 374, Sojoodi teaches that the server program is a graphical programming environment application (Fig. 1A).

Regarding claim 375, Sojoodi teaches a client graphical program which includes a graphical program creation node for creating new graphical program (col 5, lines 27-30).

Regarding claim 376, Sojoodi teaches a property node for getting property a property of the graphical program object (col 5, lines 48-52; col 6, lines 21-22).

Regarding claim 377, Sojoodi teaches an invoke node for invoking a method on the graphical program object (col 6, lines 15-19).

Regarding claim 378, Sojoodi teaches the connecting graphical program objects (col 57, lines 34-42)

Regarding claims 379-380, Sojoodi teaches a client program for creating a new graphical program which comprises means for instantiating the new graphical program (col 5, lines 28-32), means for adding an object to the new graphical program (col 38, lines 55-59), a means for getting or setting properties of the new graphical program or

Art Unit: 2173

the object (col 6, lines 21-22); a means for invoking methods on the new graphical program or the object (col 6, lines 15-17). Sojoodi does not teach creating the new program from a program without any user input specifying the nodes and interconnection of the plurality of node. However, such feature is known in the art as taught by Sullivan. Sullivan teaches steps for creating new program from a program without user input (converting a COBOL program to a Java program without user input (col 2, lines 8-27). Thus, it would have been obvious to one of ordinary skill in the art to apply Sullivan teaching of converting one program to another program without user's inputs in Sojoodi graphical program environment with the motivation being to automatically translate graphical program from one programming language to another programming language.

4. Claims 285-369 and 381 are rejected under 35 U.S.C. 103(a) as being unpatentable over McDonald et al ("McDonald", USP 5966532), Sojoodi, and Sullivan (USP 6453464)

Regarding claims 285, 333, 355, and 362, McDonald teaches the creating of a first program, when executing, programmatically creating a new graphical program (col 3, lines 61-63). McDonald differs from the claim in that McDonald does not teach the creating a new graphical program by creating graphical program nodes. However, such feature is known in the art as taught by Sojoodi. In the same field of creating graphical programs, Sojoodi teaches the creating a graphical program which creates graphical program objects (nodes) and interconnects these created graphical program objects (connects nodes) (col 5, lines 40-47). It would have been obvious to one of ordinary skill in the art, having the teaching of McDonald and Sojoodi before him at the time the

invention was made, to modify the programming system taught by McDonald to include the creating and connecting graphical program nodes taught by Sojoodi with the motivation being to enable the system to create the second graphical program from objects. McDonald and Sojoodi do not teach creating the new program from a program without any user input specifying the nodes and interconnection of the plurality of node. However, such feature is known in the art as taught by Sullivan. Sullivan teaches steps for creating new program from a program without user input (converting a COBOL program to a Java program without user input (col 2, lines 8-27). Thus, it would have been obvious to one of ordinary skill in the art to apply Sullivan's teaching of converting one program to another program without user's inputs in McDonald and Sojoodi graphical program environment with the motivation being to automatically translate graphical program from one programming language to another programming language.

Regarding claims 286 and 334, Sojoodi teaches the new graphical program comprises a diagram portion comprising a plurality of interconnected nodes and a user interface portion (col 5, lines 40-47). McDonald teaches creating the new graphical program includes creating the block diagram portion and the user interface portion (step 266 in Fig. 5).

Regarding claims 287 and 335, McDonald teaches the new graphical program comprises a data flow diagram (inherent).

Regarding claims 288 and 336, McDonald teaches the new graphical program is a virtual instrument (col. 11, lines 4-7).

Regarding claims 289 and 337, McDonald teaches the executing the first program in a first computing environment; said first computing environment is

connected to a second computing environment (inherent from step 202 in Fig. 2); said executing the first program comprises sending information from the first computing environment to the second computing environment (step 202 in Fig. 2); and the new graphical program is created in the second computing environment (col 11, lines 18-19).

Regarding claims 290 and 338, McDonald teaches the specifying the creation of the new program (col 3, lines 61-66).

Regarding claims 291 and 339, McDonald teaches the creating the new program partially based on the information received from the user (col 4, lines 11-19).

Regarding claims 292 and 340, Sojoodi teaches a system for programmatically creating a graphical program comprising a computer system including a CPU (200) and memory (204 and 206), a client program executing in the computer system, the client program performs API calls to programmatically create a graphical program (col 5, lines 28-30), a server program operable to receive the client program calls to programmatically create a graphical program and operable to perform the respective operations (col 4, lines 43-53).

Regarding claims 293 and 341, Sojoodi teaches the comprising a server program executes on another computer system which is connected to said computer system via a network (Fig. 1A).

Regarding claims 294-295 and 342-343, Sojoodi teaches the graphical program includes a block diagram, wherein the at least one graphical program object is a function node or a programmatic structure placed in the block diagram (col 16, lines 15-17).

Regarding claims 298 and 346, Sojoodi teaches a connection between input of a first graphical program object and an output of a second graphical program object (col 15, lines 26-31).

Regarding claims 299 and 347, McDonald teaches the creating one or more user interface objects wherein the one or more user interface objects perform one or more of providing input to or displaying output from the new graphical program (col 4, lines 3-7).

Regarding claims 300-301 and 348-349, McDonald teaches user interface panel (front panel objects or controls, col 4, lines 3-7) for providing input to or displaying output from the new graphical program (col 4, lines 3-7).

Regarding claims 302 and 350, McDonald teaches the obtaining a reference to a graphical program object (col 19, lines 4-6).

Regarding claims 296-297, 344-345, and 366, McDonald teaches the comprising a structure node (col 16, lines 25-27).

Regarding claims 305, 356, and 367, McDonald does not teach that the first program is a first graphical program. However, such feature is known in the art as taught by Sojoodi. Sojoodi teaches a system for accessing object capabilities in a graphical program (col 5, lines 28-32). It would have been obvious to one of ordinary skill in the art, having the teaching of McDonald and Sojoodi before him at the time the invention was made, to modify the programming system taught by McDonald to include the first graphical program taught by Sojoodi with the motivation being to enhance the visualization of the graphical programming system.



Regarding claims 306, 332, 368, and 369, Sojoodi teaches the first graphical program includes at least one object creation node for programmatically creating at least one graphical program object (col 5, lines 27-30)

Regarding claims 303, 307-308, and 351, Sojoodi teaches the first graphical program further includes a property node and the property node getting or setting a property of the graphical program object (col 5, lines 48-52). Sojoodi also teaches the object creation node outputs a reference to the graphical program object (col 5, lines 66-67).

Regarding claim 309, Sojoodi teaches an invoke node for invoking a method on the graphical program object (col 6, lines 15-19).

Regarding claims 304, 310, and 352, Sojoodi teaches the object creation node outputs a reference to the graphical program object, the invoke node receives as input the reference to the graphical program object and the invoke node invokes a method on the graphical program object specified by the reference to the graphical program object (col 5, lines 53-65).

Regarding claims 311-312, Sojoodi teaches the connecting the graphical program object to another graphical program object in the new graphical program (col 5, lines 41-47).

Regarding claim 313, Sojoodi teaches moving the graphical program object to another location in the new graphical program (col 56, lines 38-41)

Regarding claims 314-315, Sojoodi teaches displaying the object creation node; specifying a graphical program object class for the object creation node (col 13, lines 47-56)

Regarding claim 316, Sojoodi teaches the specifying position information to the object creation node (col 15, lines 32-36).

Regarding claims 317-318, Sojoodi teaches specifying owner reference information for the object creation node, the owner reference information designates an owner entity (col 15, lines 32-38).

Regarding claim 319, Sojoodi teaches the first graphical program includes at least one object creation node for programmatically creating the new graphical program (col 5, lines 27-30)

Regarding claim 320, Sojoodi teaches the displaying the graphical program creation node (col 5, lines 27-30, and specifying graphical program type (col 6, lines 12-15).

Regarding claim 321, Sojoodi teaches the graphical program creation node includes a type input (inherent).

Regarding claims 322-323, and 363-364, Sojoodi teaches the specifying a reference to a server program for the graphical program creation node (col 5, lines 8-10).

Regarding claim 324, Sojoodi teaches a server program reference input (VI Server Refnum Control 274A in Fig. 37)

Regarding claims 325 and 365, McDonald teaches the executing the first program in a first computing environment; said first computing environment is connected to a second computing environment (inherent from step 202 in Fig. 2); Sojoodi teaches the specifying a reference to a server program for the graphical program creation node (col 5, lines 8-10).

Regarding claim 326, Sojoodi teaches a graphical program creation node operable to programmatically create the new graphical program (col 5, lines 41-46) and configuring the object creation node with one or more inputs (col 5, lines 48-53).

Regarding claim 327, Sojoodi teaches the graphical program creation node outputs a reference to the new graphical program (line 66 of col 5 to line 1 of col 6).

Regarding claims 328-329, Sojoodi teaches a server reference for the graphical program creation node (VI Server Refnum Control 274A in Fig. 37).

Regarding claim 330, Sojoodi teaches specifying an object class for the object creation node (col 13, lines 47-56), specifying position information to the object creation node (col 15, lines 32-36), and specifying owner reference information for the object creation node (col 15, lines 32-38).

Regarding claim 331, Sojoodi teaches plurality of object creation nodes and graphical program objects which are connected (col 5, lines 40-47).

Regarding claims 353 and 381, McDonald teaches a first program, when executing, programmatically creating a new graphical program (col 3, lines 61-63). McDonald teaches a template graphical program wherein said creating the new graphical program comprises creating the new graphical program based on the template graphical program (col 3, lines 63-66). McDonald differs from the claim in that

Art Unit: 2173

McDonald does not teach that the first program is a first graphical program and the creating a new graphical program by creating graphical program objects and by interconnecting these created graphical program objects. However, such feature is known in the art as taught by Sojoodi. In the same field of creating graphical programs, Sojoodi teaches the creating a graphical program which creates graphical program objects (nodes) and interconnects these created graphical program objects (connects nodes) (col 5, lines 40-47). It would have been obvious to one of ordinary skill in the art, having the teaching of McDonald and Sojoodi before him at the time the invention was made, to modify the programming system taught by McDonald to include the creating and connecting graphical program nodes taught by Sojoodi with the motivation being to enable the system to create the second graphical program from objects. McDonald and Sojoodi do not teach creating the new program from a program without any user input specifying the nodes and interconnection of the plurality of node. However, such feature is known in the art as taught by Sullivan. Sullivan teaches steps for creating new program from a program without user input (converting a COBOL program to a Java program without user input (col 2, lines 8-27). Thus, it would have been obvious to one of ordinary skill in the art to apply Sullivan teaching of converting one program to another program without user's inputs in McDonald and Sojoodi graphical program environment with the motivation being to automatically translate graphical program from one programming language to another programming language.

Regarding claim 354, Sojoodi teaches the creating a first program which includes reference to an existing graphical program (col 54, lines 18-30) and modifying the

Art Unit: 2173

existing graphical program in response to the execution of the first program (col 18, lines 63-67).

Regarding claim 357, Sojoodi teaches the first program is a text-based program (col 15, line 1).

Regarding claim 358, Sojoodi teaches a method call (col 31, lines 39-41).

Regarding claim 359, Sojoodi teaches the text-based program obtains a reference to a software component (col 58, lines 42-44).

Regarding claim 360, Sojoodi teaches the software component interfaces with a server program (col 14, lines 41-46).

Regarding claim 361, Sojoodi teaches the software component is an ActiveX component (col 57, line 62).

5. Applicant's arguments filed 11/01/04 have been considered but now moot under new ground of rejection.

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kieu D. Vu.

The examiner can normally be reached on Mon - Thu from 7:00AM to 3:00PM at 571-272-4057.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca, can be reached at 571-272-4048.

The fax phone numbers for the organization where this application or proceeding is assigned are as follows:

703-872-9306

and / or:

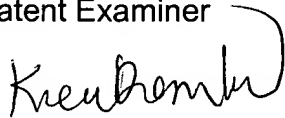
Art Unit: 2173

571-273-4057 (use this FAX #, only after approval by Examiner, for "INFORMAL" or "DRAFT" communication. Examiners may request that a formal paper / amendment be faxed directly to them on occasions).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703-305-3900).

Kieu D. Vu

Patent Examiner

A handwritten signature in black ink, appearing to read "Kieu D. Vu", with a large, sweeping flourish extending from the end of the name.